Tell me What Label Noise is

and I will Tell you how to Dispose of it

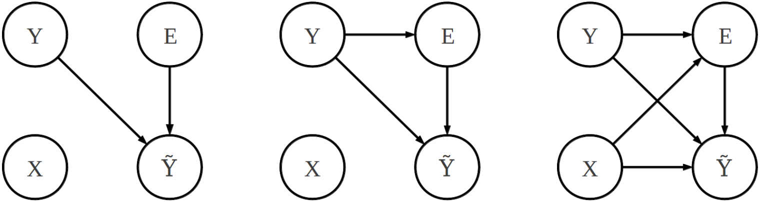
Benoît Frénay - Namur Digital Institute
PReCISE research center - UNamur

Outline of this Talk

- label noise: overview of the literature
- probabilistic models for label noise
    - HMMs for ECG segmentation
    - robust feature selection with MI
    - dealing with streaming data
    - going further with modelling
- robust maximum likelihood inference

Label Noise: Overview of the Literature

Label Noise: a Complex Phenomenon



- Frénay, B., Verleysen, M. Classification in the Presence of Label Noise: a Survey. IEEE TNN & LS, 25(5), 2014, p. 845-869.
- Frénay, B., Kabán, A. A Comprehensive Introduction to Label Noise. In Proc. ESANN, Bruges, Belgium, 23-25 April 2014, p. 667-676.

Sources and Effects of Label Noise

Label noise can come from several sources

- insufficient information provided to the expert
- errors in the expert labelling itself
- subjectivity of the labelling task
- communication/encoding problems

Label noise can have several effects

- decrease the classification performances
- increase/decrease the complexity of learned models
- pose a threat to tasks like e.g. feature selection

State-of-the-Art Methods to Deal with Label Noise

label noise-robust models rely on overfitting avoidance

- learning algorithms are seldom completely robust to label noise

data cleansing remove instances which seems to be mislabelled

- mostly based on model predictions or kNN-based methods

label noise-tolerant learning algorithms take label noise into account

- based on e.g. probabilistic models of label noise

Label Noise-Robust Models

some losses are robust to uniform label noise (Manwani and Sastry)

- theoretically robust: least-square loss $\rightarrow$ Fisher linear discriminant
- theoretically non-robust: exponential loss $\rightarrow$ AdaBoost, log loss $\rightarrow$ logistic regression, hinge loss $\rightarrow$ support vector machines

one can expect most of the recent learning algorithms in machine learning to be completely label noise-robust $\Rightarrow$ research on label noise matters!

robustness to label noise is method-specific

- boosting: AdaBoost $<$ LogitBoost / BrownBoost
- decision trees: C4.5 $<$ imprecise info-gain

there exist empirical comparisons in the literature, not easy to conclude

Label Noise-Robust Models

some losses are robust to uniform label noise (Manwani and Sastry)

- theoretically robust: least-square loss $\rightarrow$ Fisher linear discriminant
- theoretically non-robust: exponential loss $\rightarrow$ AdaBoost, log loss $\rightarrow$ logistic regression, hinge loss $\rightarrow$ support vector machines

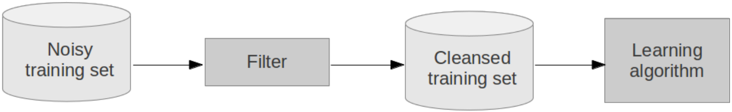
one can expect most of the recent learning algorithms in machine learning to be completely label noise-robust $\Rightarrow$ research on label noise matters!

robustness to label noise is method-specific

- boosting: AdaBoost $<$ LogitBoost / BrownBoost
- decision trees: C4.5 $<$ imprecise info-gain

there exist empirical comparisons in the literature, not easy to conclude

Data Cleansing Methods

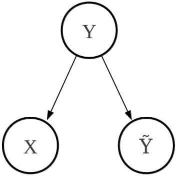


many methods to detect and remove mislabelled instances

- measures and thresholds: model complexity, entropy of $P(Y|X)\ldots$
- model prediction-based: class probabilities, voting, partition filtering
- model influence: LOO perturbed classification (LOOPC), CL-stability
- kNN: CNN, RNN, BBNR, DROP1-6, GE, Tomek links, PRISM...
- boosting: ORBoost exploits tendency of AdaBoost to overfit

Label Noise-Tolerant Methods

- $\neq$ label noise-robust $\Leftrightarrow$ fight the effects of label noise
- $\neq$ data cleansing methods $\Leftrightarrow$ no filtering of instances



- Bayesian priors and frequentist methods (e.g. Lawrence et al.)
- clustering-based: structure of data (=clusters) to model label noise
- belief functions: each instance seen as an evidence, used to robust kNN classifiers, neural networks, decision trees, boosting
- modification of SVMs, neural networks, decision trees, boosting...

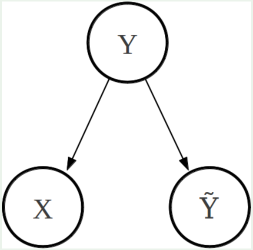
Probabilistic Models of Label Noise

Probabilistic Modelling of Lawrence et al.

p_Y = true labels Y prior

$p_{X|Y}$ = observed features X distribution

$p_{\tilde{Y}|Y}$ = observed labels $\tilde{Y}$ distribution

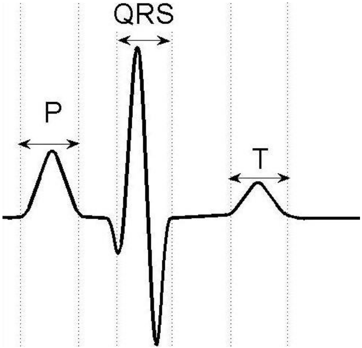


- Frénay, B., de Lannoy, G., Verleysen, M. Label noise-tolerant hidden Markov models for segmentation: application to ECGs. ECML-PKDD 2011, p. 455-470.
- Frénay, B., Doquire, G., Verleysen, M. Estimating mutual information for feature selection in the presence of label noise. CS & DA, 71, 832-848, 2014.
- Frenay, B., Hammer, B. Label-noise-tolerant classification for streaming data. IJCNN 2017, Anchorage, AK, 14-19 May 2017, p. 1748-1755.
- Bion, Q., Frénay, B. Modelling non-uniform label noise to robustify a classifier with application to neural networks. Submitted to ESANN'18 (under review).

HMMs for ECG Segmentation

What is an Electrocardiogram Signal ?

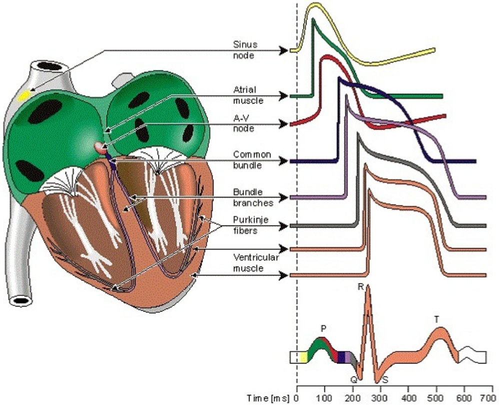
an ECG is a measure of the electrical activity of the human heart



patterns of interest: P wave, QRS complex, T wave, baseline

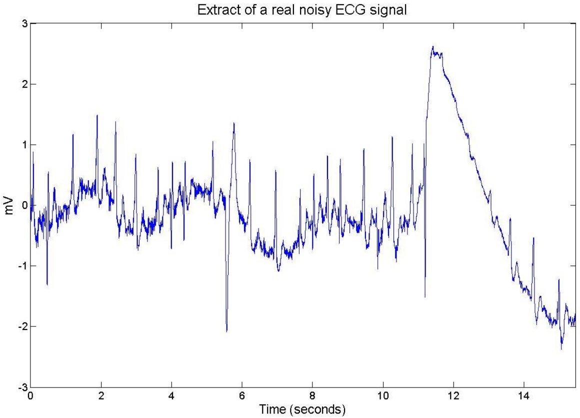
Where do ECG Signals Come from ?

an ECG results from the superposition of several signals



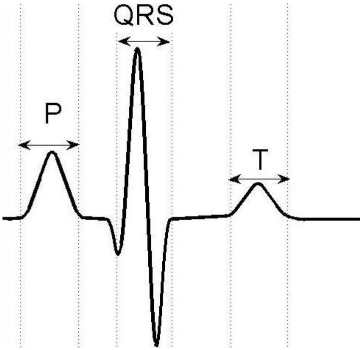
What Real-World ECG Signals Look Like

real ECGs are polluted by various sources of noise



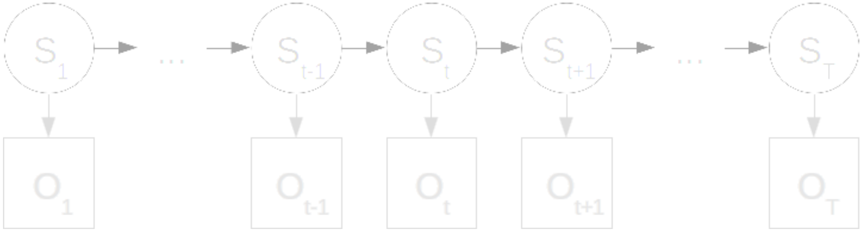
Solving an ECG Segmentation Task

- learn from a few manual segmentations from experts
- split/segment the entire ECG into patterns
- sequence modelling with hidden Markov Models (+ wavelet transform)



Hidden Markov Models in a Nutshell

hidden Markov models (HMMs) are probabilistic models of sequences



$S_1, \ldots, S_T$ is the sequence of **annotations** (ex.: state of the heart).

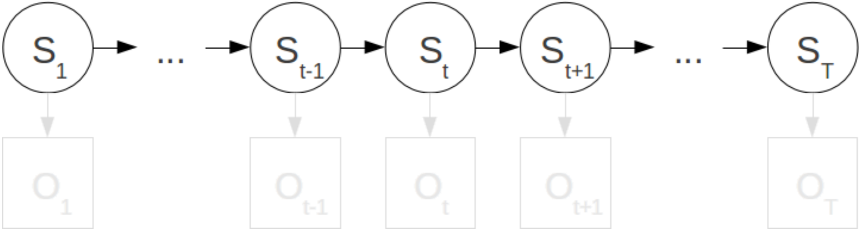
$$P(S_t = s_t | S_{t-1} = s_{t-1})$$

$O_1, \ldots, O_T$ is the sequence of **observations** (ex.: measured voltage).

$$P(O_t = o_t | S_t = s_t)$$

Hidden Markov Models in a Nutshell

hidden Markov models (HMMs) are probabilistic models of sequences



$S_1, \ldots, S_T$ is the sequence of **annotations** (ex.: state of the heart).

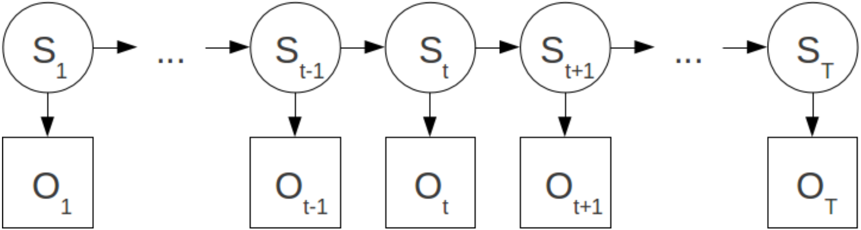
$$P(S_t = s_t | S_{t-1} = s_{t-1})$$

$O_1, \ldots, O_T$ is the sequence of **observations** (ex.: measured voltage).

$$P(O_t = o_t | S_t = s_t)$$

Hidden Markov Models in a Nutshell

hidden Markov models (HMMs) are probabilistic models of sequences



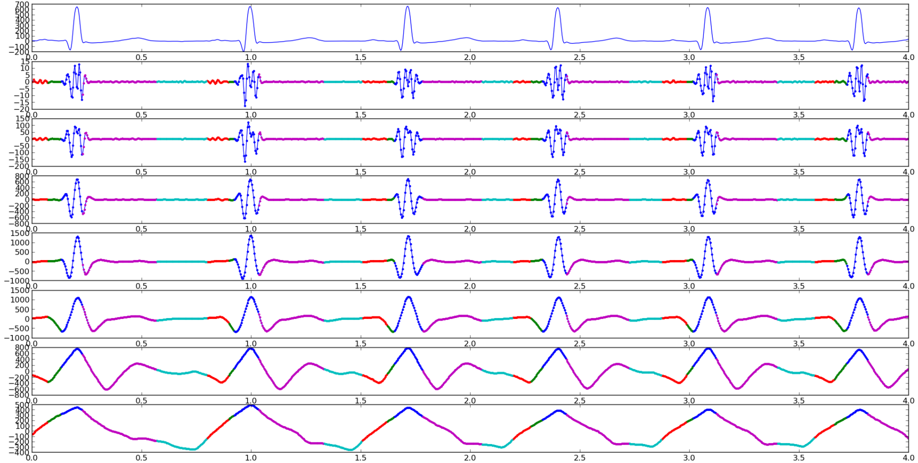
$S_1, \ldots, S_T$ is the sequence of **annotations** (ex.: state of the heart).

$$P(S_t = s_t | S_{t-1} = s_{t-1})$$

$O_1, \ldots, O_T$ is the sequence of **observations** (ex.: measured voltage).

$$P(O_t = o_t | S_t = s_t)$$

Information Extraction with Wavelet Transform



Standard Inference Algorithms for HMMs

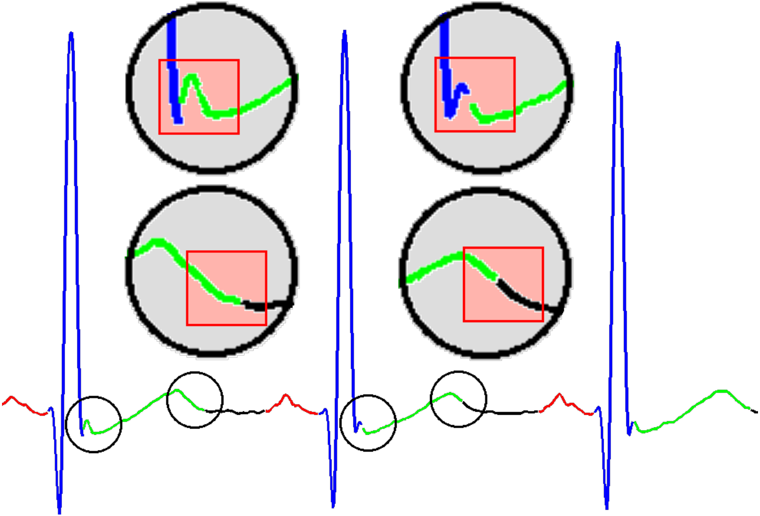
Supervised learning

- assumes the observed labels are **correct**
- **maximises** the **likelihood** $P(S, O|\Theta)$
- learns the **correct** concepts
- **sensitive** to label noise

Baum-Welch algorithm

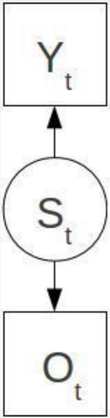
- **unsupervised**, i.e. observed labels are discarded
- iteratively (i) **label** samples and (ii) **learn** a model
- may learn **concepts** which **differs** significantly
- theoretically **insensitive** to label noise

Example of Label Noise: Electrocardiogram Signals



Label Noise Model

Modelling label noise in sequences



two distinct sequences of states:

- the observed, noisy annotations Y
- the hidden, true labels S

the annotation probability is

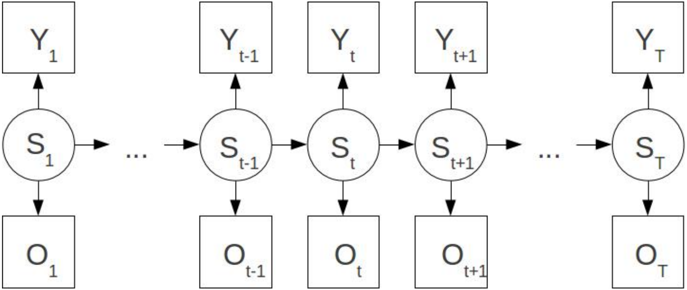
$$d_{ij} = \begin{cases} 1 - p_i & (i = j) \\ \frac{p_i}{|\mathcal{S}|-1} & (i \neq j) \end{cases}$$

where p_i is the expert error probability in i

Label Noise-Tolerant HMMs

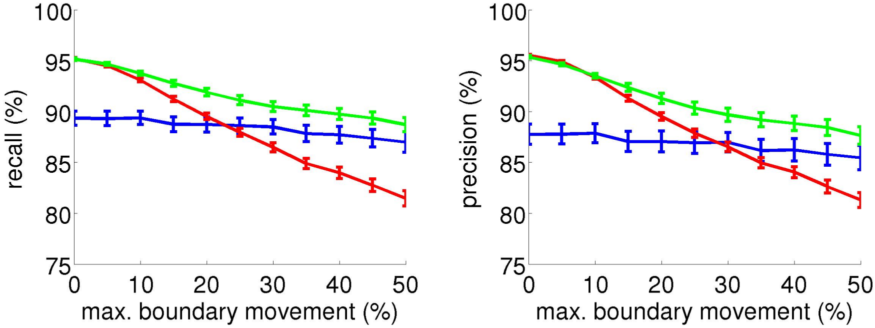
Compromise between supervised learning and Baum-Welch

- assumes the observed labels are **potentially noisy**
- **maximises** the **likelihood** $P(Y, O|\Theta) = \sum_S P(O, Y, S|\Theta)$
- learns the **correct** concepts (and error probabilities)
- **less sensitive** to label noise (can estimate label noise level)



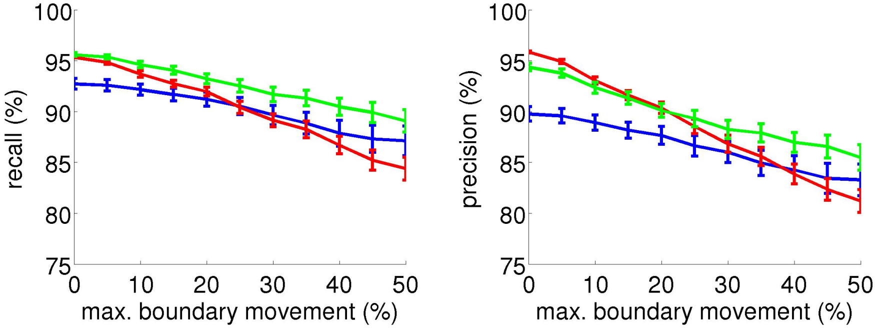
Results for Artificial ECGs

Supervised learning, Baum-Welch and label noise-tolerant.



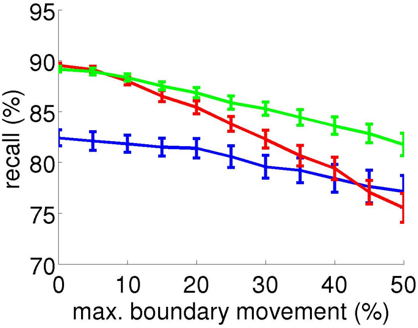
Results for Sinus ECGs

Supervised learning, **Baum-Welch** and **label noise-tolerant**.



Resuts for Arrhythmia ECGs

Supervised learning, **Baum-Welch** and **label noise-tolerant**.



Robust Feature Selection with Mutual Information

Feature Selection with Mutual Information

Problem statement

problems with high-dimensional data:

- interpretability of data
- curse of dimensionality
- concentration of distances

feature selection consists in using only a subset of the features

How to select features

mutual information (MI) assesses the quality of feature subsets:

- rigorous definition (information theory)
- interpretation in terms of uncertainty reduction
- can detect linear as well as non-linear relationships
- can be defined for multi-dimensional variables

Feature Selection with Mutual Information

Problem statement

problems with high-dimensional data:

- interpretability of data
- curse of dimensionality
- concentration of distances

feature selection consists in using only a subset of the features

How to select features

mutual information (MI) assesses the quality of feature subsets:

- rigorous definition (information theory)
- interpretation in terms of uncertainty reduction
- can detect linear as well as non-linear relationships
- can be defined for multi-dimensional variables

Label Noise-Tolerant Mutual Information Estimation

Gómez et al. propose to estimate MI (using the Kozachenko-Leonenko kNN-based entropy estimator, a.k.a. the Kraskov estimator) as

$$\begin{aligned}\hat{I}(X;Y) &= \hat{H}(X) - \sum_{y\in\mathcal{Y}} p_Y(y)\hat{H}(X|Y=y)\\ &= \psi(n) - \frac{1}{n}\sum_{y\in\mathcal{Y}} n_y\psi(n_y) + \frac{d}{n}\left[\sum_{i=1}^{n}\log\epsilon_k(i) - \sum_{y\in\mathcal{Y}}\sum_{i|y_i=y}\log\epsilon_k(i|y)\right],\end{aligned}$$

- assumption: density remains constant in a small hypersphere of diameter $\epsilon_k(i)$ containing the k nearest neighbours of the instance x_i

Solution

find hyperspheres with expected number of k instances really belonging to the target class s (with true class memberships, similar to Lawrence et al.)

Label Noise-Tolerant Mutual Information Estimation

Gómez et al. propose to estimate MI (using the Kozachenko-Leonenko kNN-based entropy estimator, a.k.a. the Kraskov estimator) as

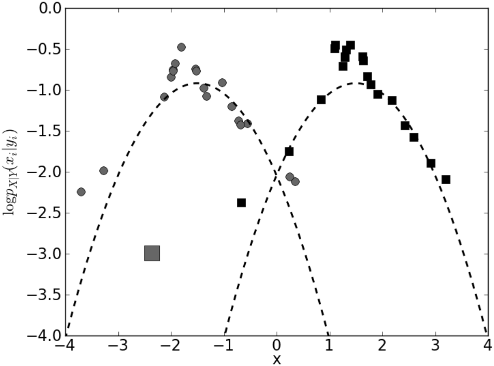
$$\begin{aligned}\hat{I}(X;Y) &= \hat{H}(X) - \sum_{y \in \mathcal{Y}} p_Y(y)\hat{H}(X|Y=y) \\ &= \psi(n) - \frac{1}{n}\sum_{y\in\mathcal{Y}} n_y \psi(n_y) + \frac{d}{n}\left[\sum_{i=1}^{n} \log \epsilon_k(i) - \sum_{y\in\mathcal{Y}}\sum_{i|y_i=y} \log \epsilon_k(i|y)\right],\end{aligned}$$

- assumption: density remains constant in a small hypersphere of diameter $\epsilon_k(i)$ containing the k nearest neighbours of the instance x_i

Solution

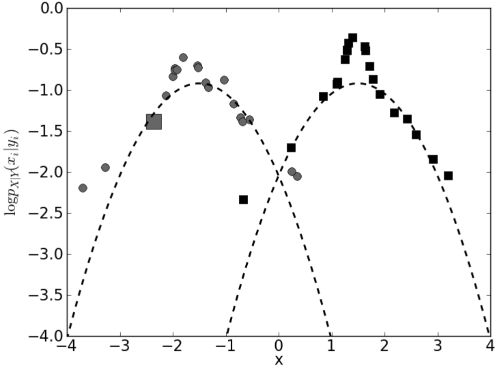
find hyperspheres with expected number of k instances really belonging to the target class s (with true class memberships, similar to Lawrence et al.)

Experimental Results for Feature Selection



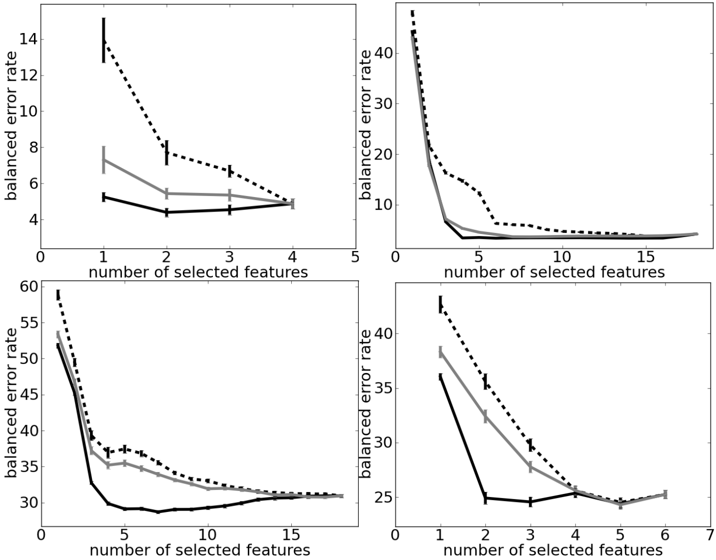
resulting estimate of MI is only $\hat{I}(X;Y) = 0.58$ instead of $\hat{I}(X;Y) = 0.63$ with clean data $\Rightarrow$ label noise-tolerant estimation is $\hat{I}(X;Y) = 0.61$

Experimental Results for Feature Selection



resulting estimate of MI is only $\hat{I}(X;Y) = 0.58$ instead of $\hat{I}(X;Y) = 0.63$ with clean data $\Rightarrow$ label noise-tolerant estimation is $\hat{I}(X;Y) = 0.61$

Experimental Results for Feature Selection



Dealing with Noisy Streaming Data

Motivation for Robust Online Learning

Context

large scale ML = deal with large (batch) or infinite (streaming) datasets

- online learning can deal with such datasets (see e.g. Botou's works)

robust classification = deal with label noise in datasets

- real-world datasets = $\pm 5\%$ labeling errors
- use of low-quality labels (crowdsourcing)

Online learning with label noise

only few online-learning approaches related to perceptron

- λ-trick modifies the adaptation criterion if previously misclassified
- α-bound does not update the weights if already misclassified α times

Motivation for Robust Online Learning

Context

large scale ML = deal with large (batch) or infinite (streaming) datasets

- online learning can deal with such datasets (see e.g. Botou's works)

robust classification = deal with label noise in datasets

- real-world datasets = $\pm 5\%$ labeling errors
- use of low-quality labels (crowdsourcing)

Online learning with label noise

only few online-learning approaches related to perceptron

- λ-trick modifies the adaptation criterion if previously misclassified
- α-bound does not update the weights if already misclassified α times

Motivation

easy-to-optimise, online learning, interpretable (controlled complexity)

RSLVQ relies on a data generating Gaussian mixture model (=prototypes)

$$p(\mathbf{x}|j) := \frac{1}{(2\pi\sigma^2)^{\frac{d}{2}}} e^{\frac{-\|\mathbf{x}-\mathbf{w}_j\|^2}{2\sigma^2}}$$

where the bandwidth σ is considered to be identical for each component.

assuming equal prior $P(j) = \frac{1}{k}$, (un)labelled instances follow

$$p(\mathbf{x}) = \frac{1}{k}\sum_{j=1}^{k} p(\mathbf{x}|j) \qquad p(\mathbf{x}, y) = \frac{1}{k}\sum_{j|c(\mathbf{w}_j)=y} p(\mathbf{x}|j)$$

Prototype-based Models: Robust Soft LVQ

Motivation

easy-to-optimise, online learning, interpretable (controlled complexity)

RSLVQ relies on a data generating Gaussian mixture model (=prototypes)

$$p(\mathbf{x}|j) := \frac{1}{(2\pi\sigma^2)^{\frac{d}{2}}} e^{\frac{-\|\mathbf{x}-\mathbf{w}_j\|^2}{2\sigma^2}}$$

where the bandwidth σ is considered to be identical for each component.

assuming equal prior $P(j) = \frac{1}{k}$, (un)labelled instances follow

$$p(\mathbf{x}) = \frac{1}{k}\sum_{j=1}^{k} p(\mathbf{x}|j) \qquad p(\mathbf{x}, y) = \frac{1}{k}\sum_{j|c(\mathbf{w}_j)=y} p(\mathbf{x}|j)$$

Prototype-based Models: Robust Soft LVQ

RSLVQ training = optimization of the conditional log likelihood

$$\sum_{i=1}^{m} \log p(y_i|\mathbf{x}_i) = \sum_{i=1}^{m} \log \frac{p(\mathbf{x}_i, y_i)}{p(\mathbf{x}_i)}$$

gradient ascent to be used in streaming scenarios ⇒ update rule

$$\Delta \mathbf{w}_j = \begin{cases} \frac{\alpha}{\sigma^2} \left(P_{y_i}(j|\mathbf{x}_i) - P(j|\mathbf{x}_i) \right) (\mathbf{x}_i - \mathbf{w}_j) & \text{if } c(\mathbf{w}_j) = y_i \\ -\frac{\alpha}{\sigma^2} P(j|\mathbf{x}_i)(\mathbf{x}_i - \mathbf{w}_j) & \text{if } c(\mathbf{w}_j) \neq y_i \end{cases}$$

where

$$P(j|\mathbf{x}_i) = \frac{p(\mathbf{x}_i|j)}{\sum_{j'=1}^{k} p(\mathbf{x}_i|j')} \qquad P_{y_i}(j|\mathbf{x}_i) = \frac{p(\mathbf{x}_i|j)}{\sum_{j'|c(\mathbf{w}_{j'})=y_i} p(\mathbf{x}_i|j')}$$

Prototype-based Models: Robust Soft LVQ

RSLVQ training = optimization of the conditional log likelihood

$$\sum_{i=1}^{m} \log p(y_i|\mathbf{x}_i) = \sum_{i=1}^{m} \log \frac{p(\mathbf{x}_i, y_i)}{p(\mathbf{x}_i)}$$

gradient ascent to be used in streaming scenarios $\Rightarrow$ update rule

$$\Delta \mathbf{w}_j = \begin{cases} \frac{\alpha}{\sigma^2}\left(P_{y_i}(j|\mathbf{x}_i) - P(j|\mathbf{x}_i)\right)(\mathbf{x}_i - \mathbf{w}_j) & \text{if } c(\mathbf{w}_j) = y_i \\ -\frac{\alpha}{\sigma^2} P(j|\mathbf{x}_i)(\mathbf{x}_i - \mathbf{w}_j) & \text{if } c(\mathbf{w}_j) \neq y_i \end{cases}$$

where

$$P(j|\mathbf{x}_i) = \frac{p(\mathbf{x}_i|j)}{\sum_{j'=1}^{k} p(\mathbf{x}_i|j')} \qquad P_{y_i}(j|\mathbf{x}_i) = \frac{p(\mathbf{x}_i|j)}{\sum_{j'|c(\mathbf{w}_{j'})=y_i} p(\mathbf{x}_i|j')}$$

Label Noise-Tolerant RSLVQ

using Lawrence and Schölkopf methodology, RSLVQ equations become

$$p(\mathbf{x}, y) = \sum_{\tilde{y} \in \mathcal{Y}} P(y|\tilde{y}) \left[\frac{1}{k} \sum_{j | c(\mathbf{w}_j) = \tilde{y}} p(\mathbf{x}|j) \right] = \frac{1}{k} \sum_{j=1}^{k} P(y|c(\mathbf{w}_j)) p(\mathbf{x}|j)$$

where $P(y|c(\mathbf{w}_j))$ = probability of observing label y if true label is $c(\mathbf{w}_j)$

online update rules become

$$\forall j \in 1 \ldots m : \Delta \mathbf{w}_j = \frac{\alpha}{\sigma^2} \left(P_{y_i}(j|\mathbf{x}_i) - P(j|\mathbf{x}_i) \right) (\mathbf{x}_i - \mathbf{w}_j)$$

where

$$P(j|\mathbf{x}_i) = \frac{p(\mathbf{x}_i|j)}{\sum_{j'=1}^{k} p(\mathbf{x}_i|j')} \qquad P_{y_i}(j|\mathbf{x}_i) = \frac{P(y_i|c(\mathbf{w}_j)) p(\mathbf{x}_i|j)}{\sum_{j'=1}^{k} P(y_i|c(\mathbf{w}_{j'})) p(\mathbf{x}_i|j')}$$

Label Noise-Tolerant RSLVQ

using Lawrence and Schölkopf methodology, RSLVQ equations become

$$p(\mathbf{x}, y) = \sum_{\tilde{y} \in \mathcal{Y}} P(y|\tilde{y}) \left[\frac{1}{k} \sum_{j|c(\mathbf{w}_j)=\tilde{y}} p(\mathbf{x}|j) \right] = \frac{1}{k} \sum_{j=1}^{k} P(y|c(\mathbf{w}_j)) p(\mathbf{x}|j)$$

where $P(y|c(\mathbf{w}_j))$ = probability of observing label y if true label is $c(\mathbf{w}_j)$

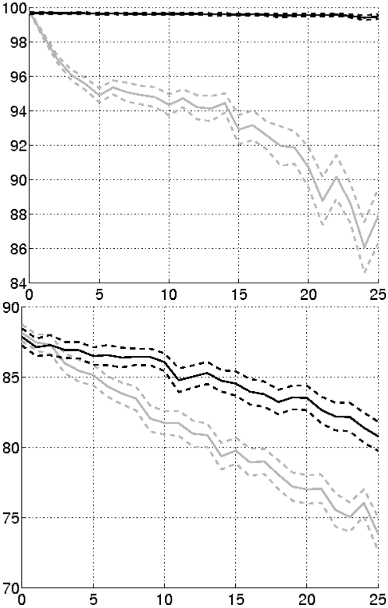
online update rules become

$$\forall j \in 1 \dots m : \Delta \mathbf{w}_j = \frac{\alpha}{\sigma^2} \left(P_{y_i}(j|\mathbf{x}_i) - P(j|\mathbf{x}_i) \right) (\mathbf{x}_i - \mathbf{w}_j)$$

where

$$P(j|\mathbf{x}_i) = \frac{p(\mathbf{x}_i|j)}{\sum_{j'=1}^{k} p(\mathbf{x}_i|j')} \qquad P_{y_i}(j|\mathbf{x}_i) = \frac{P(y_i|c(\mathbf{w}_j)) p(\mathbf{x}_i|j)}{\sum_{j'=1}^{k} P(y_i|c(\mathbf{w}_{j'})) p(\mathbf{x}_i|j')}$$

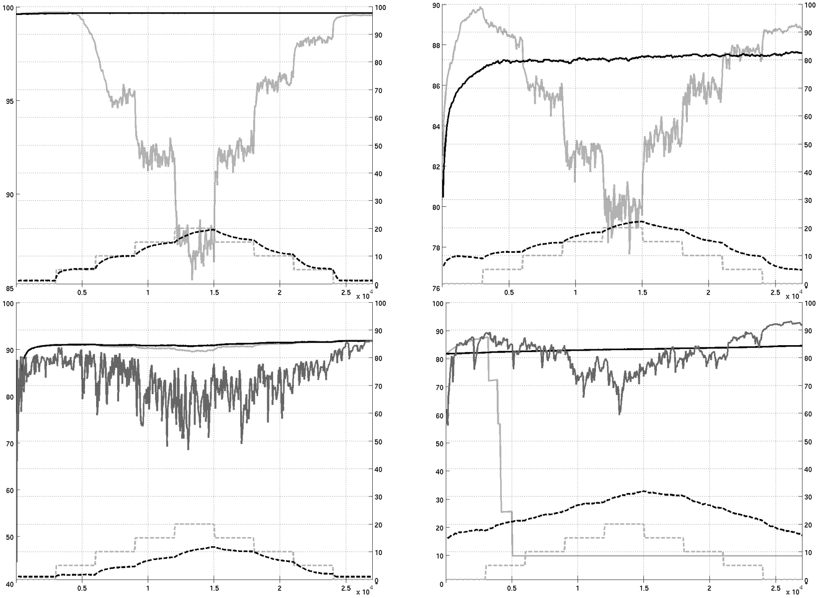
Results in batch setting



Results in batch setting

name	10% of label noise			20% of noise of noise		
	RSLVQ	LNT	p-value	RSLVQ	LNT	p-value
Bupa	66.50	**68.43**	0.00	63.76	65.06	0.09
Haberman	72.64	**73.91**	0.03	70.35	**73.02**	0.00
Ionosphere	81.67	**86.41**	0.00	76.05	**82.38**	0.00
Mammographis	81.65	81.76	0.75	80.28	80.87	0.13
Optdigits	94.81	**99.69**	0.00	90.97	**99.60**	0.00
Parkinsons	79.40	80.55	0.20	71.67	**77.22**	0.00
Pima	73.88	73.83	0.90	71.50	**72.44**	0.04
Sonar	73.19	**77.39**	0.00	65.05	**73.73**	0.00
Votes	89.49	**94.09**	0.00	85.24	**92.04**	0.00
Wdbc	90.02	**96.06**	0.00	86.02	**95.14**	0.00
Iris	94.42	94.98	0.26	90.60	**94.02**	0.00
Glass	71.00	**74.88**	0.00	67.81	**73.90**	0.00
Wine	87.08	**96.72**	0.00	78.15	**95.19**	0.00
Vertebral	78.99	**81.03**	0.00	75.66	**79.60**	0.00
Vehicle	77.93	77.64	0.47	75.14	75.15	0.99
Ecoli	81.63	**84.07**	0.00	78.43	**83.23**	0.00
Breast tissue	59.90	60.65	0.56	54.65	**59.61**	0.00

Results in streaming setting

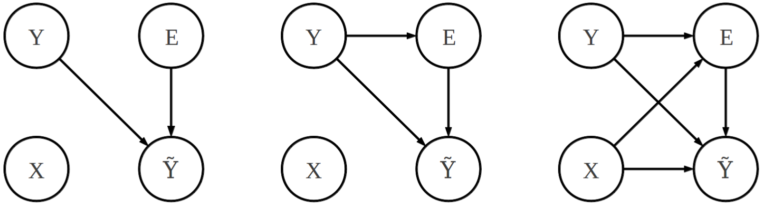


Results in streaming setting

name	10% of label noise		20% of label noise		10% of label noise	
	RSLVQ	LNT	RSLVQ	LNT	RSLVQ	LNT
Bupa	68.3 (3.5)	**70.1**(2.6)	66.6 (4.2)	**69.6**(2.6)	67.2 (3.7)	**69.7**(2.3)
Haberman	73.0 (1.9)	**74.7**(0.7)	71.8 (2.5)	**74.5**(0.8)	72.6 (1.9)	**74.5**(0.7)
Ionosphere	85.8 (3.9)	**87.2**(0.5)	79.9 (7.4)	**87.3**(0.4)	85.7 (3.8)	**87.5**(0.4)
Mammo.	81.3 (2.0)	81.3(0.4)	78.1 (4.7)	**81.5**(0.4)	80.8 (2.0)	**81.6**(0.5)
Optdigits	95.6 (2.8)	**99.7**(0.0)	87.3 (7.4)	**99.7**(0.0)	95.9 (3.0)	**99.7**(0.0)
Parkinsons	**87.3** (3.8)	77.3(1.2)	**83.2** (7.3)	80.4(1.2)	**88.2** (3.6)	83.3(1.2)
Pima	73.1 (2.6)	**76.1**(1.1)	71.5 (3.2)	**75.6**(1.2)	73.3 (2.5)	**75.6**(1.1)
Sonar	77.8 (4.8)	**80.9**(1.4)	72.3 (6.9)	**81.2**(1.0)	77.3 (4.6)	**81.4**(0.9)
Votes	**94.1** (1.6)	92.5(0.5)	91.1 (3.9)	**93.5**(0.3)	**94.8** (1.4)	94.1(0.4)
Wdbc	87.4 (4.0)	**94.3**(0.2)	83.8 (8.0)	**94.7**(0.1)	91.1 (3.3)	**94.9**(0.2)
Iris	91.4 (3.4)	**95.2**(0.4)	93.1 (3.3)	**95.2**(0.4)	95.7 (1.6)	95.2(0.4)
Glass	74.4 (3.8)	**76.4**(2.5)	70.0 (4.9)	**76.5**(2.0)	72.8 (3.5)	**76.8**(1.9)
Wine	96.1 (1.9)	**97.3**(0.2)	93.6 (3.4)	**97.4**(0.1)	96.4 (1.9)	**97.4**(0.1)
Vertebral	**81.4** (2.6)	79.2(1.0)	79.6 (3.7)	79.6(0.9)	**81.7** (2.5)	80.4(1.1)
Waveform	82.5 (1.4)	**85.6**(0.5)	78.1 (2.3)	**85.6**(0.4)	81.9 (1.4)	**85.5**(0.4)
Vehicle	**76.8** (2.7)	69.0(2.2)	**75.8** (3.5)	73.9(2.0)	**77.9** (2.8)	76.5(2.0)
Wall robot	**74.9** (1.8)	74.5(1.4)	74.0 (2.3)	**77.4**(0.7)	77.3 (1.7)	**78.7**(0.7)
Ecoli	83.0 (2.0)	**84.9**(0.6)	82.9 (2.4)	**84.7**(0.4)	85.7 (1.7)	84.9(0.4)
Breast tissue	37.0 (2.1)	**61.4**(1.2)	35.4 (2.3)	**61.7**(0.9)	35.5 (2.0)	**61.7**(0.7)
	Projectron++	LNT	Projectron++	LNT	Projectron++	LNT
a9a	73.4 (9.4)	**81.7**(0.7)	68.0 (16.2)	**83.1**(0.2)	73.7 (8.8)	**83.7**(0.2)
IJCNN1	83.0 (4.6)	**90.9**(0.2)	76.0 (5.7)	**90.8**(0.2)	82.1 (3.9)	**91.4**(0.2)
MNIST	**83.3** (3.5)	82.8(0.2)	74.4 (4.3)	**83.4**(0.1)	82.3 (3.1)	**83.9**(0.1)

Going Further with Modelling

Statistical Taxonomy of Label Noise (inspired by Schafer)



most works consider that label noise affects instances with no distinction

- in specific cases, empirical evidence was given that more difficult samples are labelled randomly (e.g. in text entailment)
- it seems natural to expect less reliable labels in regions of low density

Non-uniform Label Noise in the Literature

Lachenbruch/Chhikara models of label noise

probability of misallocation $g_y(z)$ for LDA is defined w.r.t a z-axis which passes through the center of both classes s.t. each center is at $z = \pm\frac{\Delta}{2}$

- random misallocation: $g_y(z) = \alpha_y$ is constant for each class
- truncated label noise: $g(z)$ is zero as long as the instance is close enough to the mean of its class, then equal to a small constant
- exponential model:

$$g_y(z) = \begin{cases} 0 & \text{if } z \leq -\frac{\Delta}{2} \\ 1 - \exp\left(-\frac{1}{2}k_y(z + \frac{\Delta}{2})^2\right) & \text{if } z > -\frac{\Delta}{2} \end{cases}$$

Non-uniform Label Noise in the Literature

non-uniform label noise is considered in much less works

- experiments rely on simple models (e.g. Lachenbruch/Chhikara's or Sastry's quadrant-dependent probability of mislabelling)
- there are (up to our knowledge) almost no empirical evidences/studies on the characteristics of real non-uniform label noise

Call for discussions

It would be very interesting to obtain more real-world datasets where mislabelled instances are clearly identified. Also, an important open research problem is to find what the characteristics of real-world label noise are. It is not yet clear if and when (non-)uniform label noise is realistic.

Non-uniform Label Noise in the Literature

non-uniform label noise is considered in much less works

- experiments rely on simple models (e.g. Lachenbruch/Chhikara's or Sastry's quadrant-dependent probability of mislabelling)
- there are (up to our knowledge) almost no empirical evidences/studies on the characteristics of real non-uniform label noise

Call for discussions

It would be very interesting to obtain more real-world datasets where mislabelled instances are clearly identified. Also, an important open research problem is to find what the characteristics of real-world label noise are. It is not yet clear if and when (non-)uniform label noise is realistic.

What if Label Noise $\perp$ Classification Task?

Work in Progress

Quentin Bion, a student from Nancy (École des Mines, Ing. Mathématique) made an internship in our lab and worked on non-uniform label noise

- results have been submitted to the ESANN'18 conference
- he studied how to combine simple non-uniform models of label noise with complex classification models using generic mechanisms
- gain = power of up-to-date classifiers + interpretability/transparency of simple non-uniform models of label noise = best of both worlds

Robust Maximum Likelihood Inference

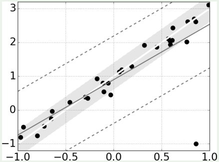
What is the common point of…

- linear regression
- LS-SVMs
- logistic regression
- principal component analysis

Answer

- common methods in machine learning
- can interpreted in probabilistic terms
- sensitive to outliers (but can be robustified)

What is the common point of...

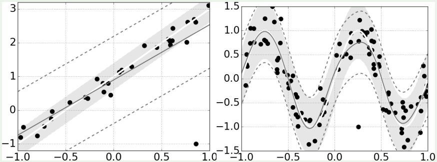


- linear regression
- LS-SVMs
- logistic regression
- principal component analysis

Answer

- common methods in machine learning
- can interpreted in probabilistic terms
- sensitive to outliers (but can be robustified)

What is the common point of…

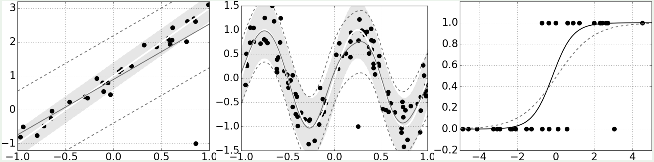


- linear regression
- LS-SVMs
- logistic regression
- principal component analysis

Answer

- common methods in machine learning
- can interpreted in probabilistic terms
- sensitive to outliers (but can be robustified)

What is the common point of. . .

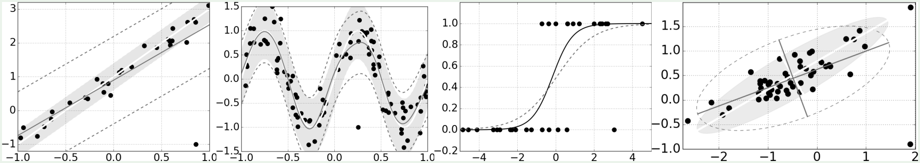


- linear regression
- LS-SVMs
- logistic regression
- principal component analysis

Answer

- common methods in machine learning
- can interpreted in probabilistic terms
- sensitive to outliers (but can be robustified)

What is the common point of. . .

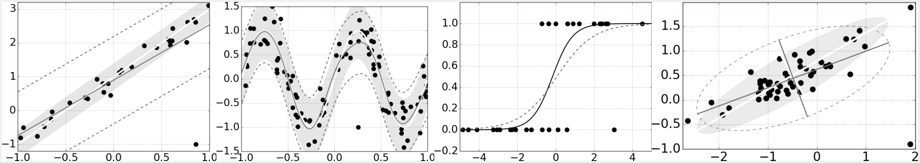


- linear regression
- LS-SVMs
- logistic regression
- principal component analysis

Answer

- common methods in machine learning
- can interpreted in probabilistic terms
- sensitive to outliers (but can be robustified)

What is the common point of...



- linear regression
- LS-SVMs
- logistic regression
- principal component analysis

Answer

- common methods in machine learning
- can interpreted in probabilistic terms
- sensitive to outliers (but can be robustified)

Everything Wrong with Maximum Likelihood Inference

What is maximum likelihood inference?

- maximise loglikelihood $\mathcal{L}(\theta; \mathbf{x}) = \log p(x_1, .., x_n) = \sum_{i=1}^{n} \log p(x_i|\theta)$
- minimise KL divergence of empirical vs. parametric distribution

$$D_{\mathrm{KL}}(\text{empirical distr.} \| \text{parametric distr.}) = -\sum_{i=1}^{n} \log p(x_i|\theta) + \text{const.}$$

Why is it sensitive to outliers?

abnormally frequent data / outliers = too frequent observations

- the reference distribution in D_{KL} is the empirical one
- inference is biased towards models supporting outliers
- otherwise, D_{KL} is too large because of low probability for outliers

Everything Wrong with Maximum Likelihood Inference

What is maximum likelihood inference?

- maximise loglikelihood $\mathcal{L}(\boldsymbol{\theta}; \mathbf{x}) = \log p(x_1, .., x_n) = \sum_{i=1}^{n} \log p(x_i|\boldsymbol{\theta})$
- minimise KL divergence of empirical vs. parametric distribution

$$D_{\mathrm{KL}}(\text{empirical distr.} \| \text{parametric distr.}) = -\sum_{i=1}^{n} \log p(x_i|\theta) + \text{const.}$$

Why is it sensitive to outliers?

abnormally frequent data / outliers = too frequent observations

- the reference distribution in D_{KL} is the empirical one
- inference is biased towards models supporting outliers
- otherwise, D_{KL} is too large because of low probability for outliers

Everything Wrong with Maximum Likelihood Inference

What is maximum likelihood inference?

- maximise loglikelihood $\mathcal{L}(\boldsymbol{\theta}; \mathbf{x}) = \log p(x_1, .., x_n) = \sum_{i=1}^{n} \log p(x_i|\boldsymbol{\theta})$
- minimise KL divergence of empirical vs. parametric distribution

$$D_{\mathrm{KL}}(\text{empirical distr.} \| \text{parametric distr.}) = -\sum_{i=1}^{n} \log p(x_i|\boldsymbol{\theta}) + \text{const.}$$

Why is it sensitive to outliers?

abnormally frequent data / outliers = too frequent observations

- the reference distribution in D_{KL} is the empirical one
- inference is biased towards models supporting outliers
- otherwise, D_{KL} is too large because of low probability for outliers

Everything Wrong with Maximum Likelihood Inference

What is maximum likelihood inference?

- maximise loglikelihood $\mathcal{L}(\boldsymbol{\theta}; \mathbf{x}) = \log p(x_1, .., x_n) = \sum_{i=1}^{n} \log p(x_i|\boldsymbol{\theta})$
- minimise KL divergence of empirical vs. parametric distribution

$$D_{\mathrm{KL}}(\text{empirical distr.}\|\text{parametric distr.}) = -\sum_{i=1}^{n} \log p(x_i|\boldsymbol{\theta}) + \text{const.}$$

Why is it sensitive to outliers?

abnormally frequent data / outliers = too frequent observations

- the reference distribution in D_{KL} is the empirical one
- inference is biased towards models supporting outliers
- otherwise, D_{KL} is too large because of low probability for outliers

Everything Wrong with Maximum Likelihood Inference

What is maximum likelihood inference?

- maximise loglikelihood $\mathcal{L}(\boldsymbol{\theta}; \mathbf{x}) = \log p(x_1, .., x_n) = \sum_{i=1}^{n} \log p(x_i|\boldsymbol{\theta})$
- minimise KL divergence of empirical vs. parametric distribution

$$D_{\mathrm{KL}}(\text{empirical distr.}\|\text{parametric distr.}) = -\sum_{i=1}^{n} \log p(x_i|\boldsymbol{\theta}) + \text{const.}$$

Why is it sensitive to outliers?

abnormally frequent data / outliers = too frequent observations

- the reference distribution in D_{KL} is the empirical one
- inference is biased towards models supporting outliers
- otherwise, D_{KL} is too large because of low probability for outliers

Pointwise Probability Reinforcements

Idea: provide an alternative to deal with outliers

good distributions cannot explain outliers $\Rightarrow$ low probability

- consequence: not-so-good distributions are enforced
- our solution: provide another mechanism to deal with outliers

Reinforced loglikelihood

x_i is given a PPR $r(x_i) \geq 0$ as reinforcement to the probability $p(x_i|\boldsymbol{\theta})$

$$\mathcal{L}(\boldsymbol{\theta}; \mathrm{x}, r) = \sum_{i=1}^{n} \log \left[p(x_i|\boldsymbol{\theta}) + r(x_i) \right]$$

reinforced maximum likelihood inference:

- PPRs $r(x_i) \approx 0$ if $x_i =$ clean $\Rightarrow$ x_i impacts the inference of $\hat{\boldsymbol{\theta}}$
- PPRs $r(x_i) \gg 0$ if $x_i =$ outlier $\Rightarrow$ x_i ignored by inference of $\hat{\boldsymbol{\theta}}$

Pointwise Probability Reinforcements

Idea: provide an alternative to deal with outliers

good distributions cannot explain outliers $\Rightarrow$ low probability

- consequence: not-so-good distributions are enforced
- our solution: provide another mechanism to deal with outliers

Reinforced loglikelihood

x_i is given a PPR $r(x_i) \geq 0$ as reinforcement to the probability $p\left(x_i|\boldsymbol{\theta}\right)$

$$\mathcal{L}\left(\boldsymbol{\theta}; \mathbf{x}, r\right) = \sum_{i=1}^{n} \log\left[p\left(x_i|\boldsymbol{\theta}\right) + r(x_i)\right]$$

reinforced maximum likelihood inference:

- PPRs $r(x_i) \approx 0$ if $x_i =$ clean $\Rightarrow$ x_i impacts the inference of $\hat{\boldsymbol{\theta}}$
- PPRs $r(x_i) \gg 0$ if $x_i =$ outlier $\Rightarrow$ x_i ignored by inference of $\hat{\boldsymbol{\theta}}$

Control of PPRS through Regularisation

Keeping the PPRs under Control

PPRs cannot be allowed to take any arbitrary positive values

- non-parametric approach: reinforcement $r(\mathbf{x}_i) = r_i$ for each $\mathbf{x}_i$

$$\mathcal{L}_{\Omega}\left(\boldsymbol{\theta}; \mathbf{X}, \mathbf{r}\right) = \sum_{i=1}^{n} \log\left[p\left(\mathbf{x}_i | \boldsymbol{\theta}\right) + r_i\right] - \alpha\Omega(\mathbf{r})$$

- no prior knowledge on outliers (e.g. uniform distribution of outliers or mislabelling probability w.r.t. distance to the classification boundary)
- $\Omega(\mathbf{r})$ = penalisation term (e.g. to allow only a few non-zero PPRs)
- α = focus on fitting the model vs. considering data as outliers

Control of PPRS through Regularisation

Keeping the PPRs under Control

PPRs cannot be allowed to take any arbitrary positive values

- non-parametric approach: reinforcement $r(\mathbf{x}_i) = r_i$ for each $\mathbf{x}_i$

$$\mathcal{L}_\Omega\left(\boldsymbol{\theta}; \mathbf{X}, \mathbf{r}\right) = \sum_{i=1}^{n} \log\left[p\left(\mathbf{x}_i|\boldsymbol{\theta}\right) + r_i\right] - \alpha\Omega(\mathbf{r})$$

- no prior knowledge on outliers (e.g. uniform distribution of outliers or mislabelling probability w.r.t. distance to the classification boundary)
- $\Omega(\mathbf{r})$ = penalisation term (e.g. to allow only a few non-zero PPRs)
- α = focus on fitting the model vs. considering data as outliers

Control of PPRS through Regularisation

Keeping the PPRs under Control

PPRs cannot be allowed to take any arbitrary positive values

- non-parametric approach: reinforcement $r(\mathbf{x}_i) = r_i$ for each $\mathbf{x}_i$

$$\mathcal{L}_\Omega\left(\boldsymbol{\theta}; \mathbf{X}, \mathbf{r}\right) = \sum_{i=1}^{n} \log\left[p\left(\mathbf{x}_i | \boldsymbol{\theta}\right) + r_i\right] - \alpha\Omega(\mathbf{r})$$

- no prior knowledge on outliers (e.g. uniform distribution of outliers or mislabelling probability w.r.t. distance to the classification boundary)
- $\Omega(\mathbf{r})$ = penalisation term (e.g. to allow only a few non-zero PPRs)
- α = focus on fitting the model vs. considering data as outliers

Control of PPRS through Regularisation

Keeping the PPRs under Control

PPRs cannot be allowed to take any arbitrary positive values

- non-parametric approach: reinforcement $r(\mathbf{x}_i) = r_i$ for each $\mathbf{x}_i$

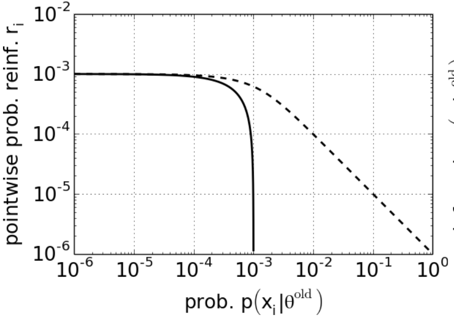
$$\mathcal{L}_\Omega\left(\boldsymbol{\theta}; \mathbf{X}, \mathbf{r}\right) = \sum_{i=1}^{n} \log\left[p\left(\mathbf{x}_i|\boldsymbol{\theta}\right) + r_i\right] - \alpha\Omega(\mathbf{r})$$

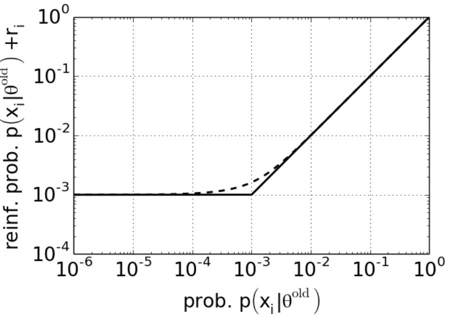
- no prior knowledge on outliers (e.g. uniform distribution of outliers or mislabelling probability w.r.t. distance to the classification boundary)
- $\Omega(\mathbf{r})$ = penalisation term (e.g. to allow only a few non-zero PPRs)
- α = focus on fitting the model vs. considering data as outliers

Control of PPRS through Regularisation

theoretical guarantees and closed form expressions exist for some $\Omega(\mathbf{r})$

- L1 penalisation $\Omega(\mathbf{r}) = \sum_{i=1}^{n} r_i$ shrinks (sparse) PPRs towards zero
- L2 regularisation $\Omega(\mathbf{r}) = \frac{1}{2}\sum_{i=1}^{n} r_i^2$ provides smoother solutions



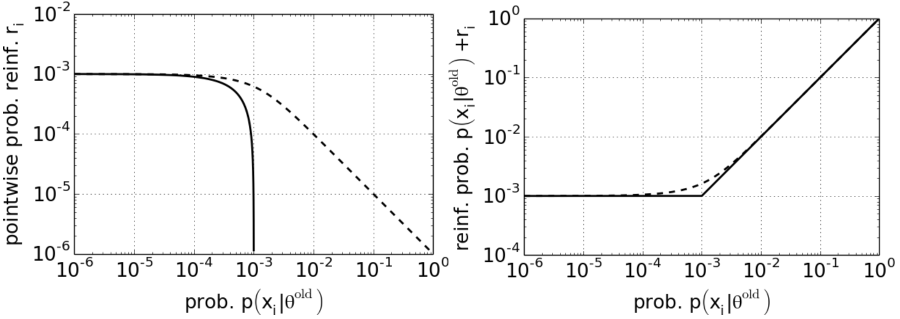


all details in Frénay, B., Verleysen, M. Pointwise Probability Reinforcements for Robust Statistical Inference. Neural networks, 50, 124-141, 2014. short version: "Robustifying Maximum Likelihood Inference" at BENELEARN'16

Control of PPRS through Regularisation

theoretical guarantees and closed form expressions exist for some $\Omega(\mathbf{r})$

- L1 penalisation $\Omega(\mathbf{r}) = \sum_{i=1}^{n} r_i$ shrinks (sparse) PPRs towards zero
- L2 regularisation $\Omega(\mathbf{r}) = \frac{1}{2}\sum_{i=1}^{n} r_i^2$ provides smoother solutions



all details in Frénay, B., Verleysen, M. Pointwise Probability Reinforcements for Robust Statistical Inference. Neural networks, 50, 124-141, 2014. short version: "Robustifying Maximum Likelihood Inference" at BENELEARN'16

Iterative Method for Reinforced Inference

no closed-form solution for penalised reinforced log-likelihood... but

$$\sum_{i=1}^{n} \log \left[p\left(\mathrm{x}_i | \boldsymbol{\theta}\right) + r_i \right] \geq \sum_{i=1}^{n} w_i \log p\left(\mathrm{x}_i | \boldsymbol{\theta}\right) + const.$$

where $\boldsymbol{\theta}^{\mathrm{old}}$ = current estimate and $w_i = p\left(\mathrm{x}_i | \boldsymbol{\theta}^{\mathrm{old}}\right) / p\left(\mathrm{x}_i | \boldsymbol{\theta}^{\mathrm{old}}\right) + r_i$

EM-like algorithm

initialise θ, then loop over

1. compute PPRs with L1 or L2 regularisation $\Rightarrow$ model-independent
2. compute instance weights with PPRs $\Rightarrow$ model-independent
3. maximise weighted log-likelihood to update θ $\Rightarrow$ model-specific

Iterative Method for Reinforced Inference

no closed-form solution for penalised reinforced log-likelihood... but

$$\sum_{i=1}^{n} \log\left[p\left(\mathbf{x}_i|\boldsymbol{\theta}\right)+r_i\right] \geq \sum_{i=1}^{n} w_i \log p\left(\mathbf{x}_i|\boldsymbol{\theta}\right) + const.$$

where $\boldsymbol{\theta}^{\text{old}}$ = current estimate and $w_i = p\left(\mathbf{x}_i|\boldsymbol{\theta}^{\text{old}}\right) / p\left(\mathbf{x}_i|\boldsymbol{\theta}^{\text{old}}\right) + r_i$

EM-like algorithm

initialise $\boldsymbol{\theta}$, then loop over

1. compute PPRs with L1 or L2 regularisation $\Rightarrow$ model-independent
2. compute instance weights with PPRs $\Rightarrow$ model-independent
3. maximise weighted log-likelihood to update $\boldsymbol{\theta}$ $\Rightarrow$ model-specific

Iterative Method for Reinforced Inference

no closed-form solution for penalised reinforced log-likelihood... but

$$\sum_{i=1}^{n} \log \left[p\left(\mathbf{x}_i | \boldsymbol{\theta}\right) + r_i \right] \geq \sum_{i=1}^{n} w_i \log p\left(\mathbf{x}_i | \boldsymbol{\theta}\right) + const.$$

where $\boldsymbol{\theta}^{\text{old}}$ = current estimate and $w_i = p\left(\mathbf{x}_i | \boldsymbol{\theta}^{\text{old}}\right) / p\left(\mathbf{x}_i | \boldsymbol{\theta}^{\text{old}}\right) + r_i$

EM-like algorithm

initialise $\boldsymbol{\theta}$, then loop over

1. compute PPRs with L1 or L2 regularisation $\Rightarrow$ model-independent
2. compute instance weights with PPRs $\Rightarrow$ model-independent
3. maximise weighted log-likelihood to update $\boldsymbol{\theta}$ $\Rightarrow$ model-specific

Iterative Method for Reinforced Inference

no closed-form solution for penalised reinforced log-likelihood... but

$$\sum_{i=1}^{n} \log \left[p\left(\mathrm{x}_i | \boldsymbol{\theta}\right) + r_i \right] \geq \sum_{i=1}^{n} w_i \log p\left(\mathrm{x}_i | \boldsymbol{\theta}\right) + const.$$

where $\boldsymbol{\theta}^{\text{old}}$ = current estimate and $w_i = p\left(\mathrm{x}_i | \boldsymbol{\theta}^{\text{old}}\right) / p\left(\mathrm{x}_i | \boldsymbol{\theta}^{\text{old}}\right) + r_i$

EM-like algorithm

initialise $\boldsymbol{\theta}$, then loop over

1. compute PPRs with L1 or L2 regularisation $\Rightarrow$ model-independent
2. compute instance weights with PPRs $\Rightarrow$ model-independent
3. maximise weighted log-likelihood to update $\boldsymbol{\theta}$ $\Rightarrow$ model-specific

Iterative Method for Reinforced Inference

no closed-form solution for penalised reinforced log-likelihood... but

$$\sum_{i=1}^{n} \log \left[p\left(\mathrm{x}_i | \boldsymbol{\theta}\right) + r_i \right] \geq \sum_{i=1}^{n} w_i \log p\left(\mathrm{x}_i | \boldsymbol{\theta}\right) + const.$$

where $\boldsymbol{\theta}^{\mathrm{old}}$ = current estimate and $w_i = p\left(\mathrm{x}_i | \boldsymbol{\theta}^{\mathrm{old}}\right) / p\left(\mathrm{x}_i | \boldsymbol{\theta}^{\mathrm{old}}\right) + r_i$

EM-like algorithm

initialise $\boldsymbol{\theta}$, then loop over

1. compute PPRs with L1 or L2 regularisation $\Rightarrow$ model-independent
2. compute instance weights with PPRs $\Rightarrow$ model-independent
3. maximise weighted log-likelihood to update $\boldsymbol{\theta}$ $\Rightarrow$ model-specific

Illustration: Linear Regression / OLS

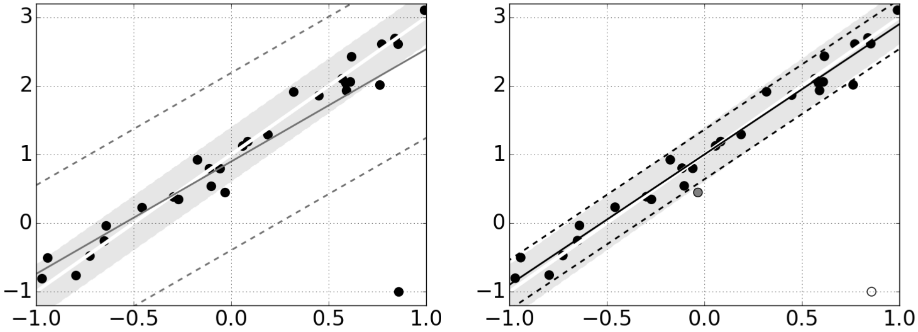
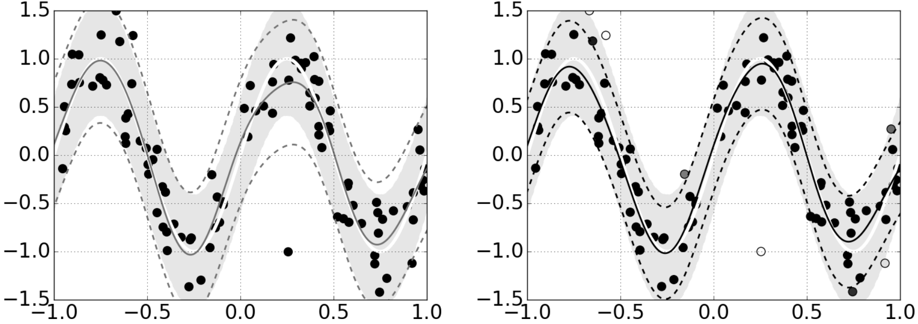
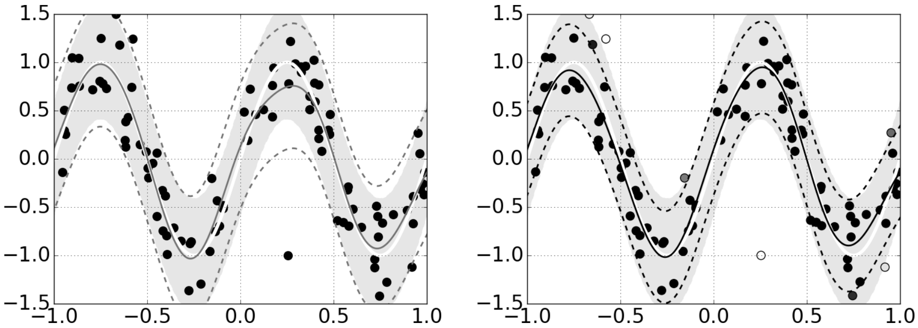


Illustration: LS-SVMs



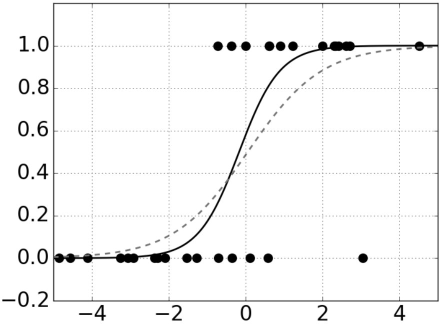
comparison to Suykens et al. (2002) on 22 datasets ⇒ see full paper

Illustration: LS-SVMs



comparison to Suykens et al. (2002) on 22 datasets $\Rightarrow$ see full paper

Illustration: Logistic Regression



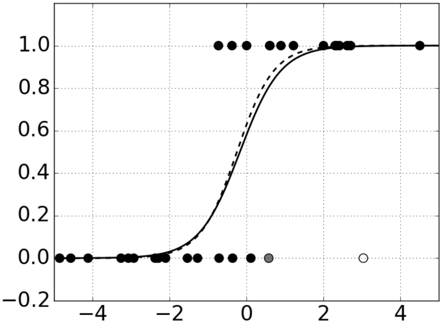
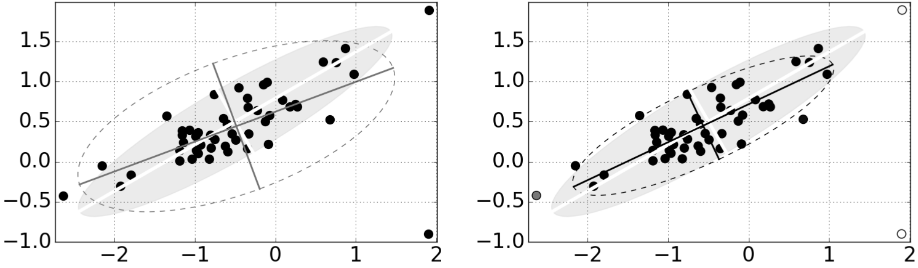


Illustration: Principal Component Analysis



Conclusion

Take Home Messages

Probabilistic approaches

probabilistic modelling of label noise is a powerful approach

- easily plugged in various problems/models
- can be used to provide feedback to users
- can embed prior knowledge on label noise

Pointwise probability reinforcements

generic approach to robustify maximum likelihood inference

- many models can be formulated in probabilistic terms
- no parametric assumption $\rightarrow$ could deal with non-uniform noise
- easy to implement if weights can be enforced on instances
- further work: more complex models + noise level estimation

try on your own favorite probabilistic model (and let us know!)

Take Home Messages

Probabilistic approaches

probabilistic modelling of label noise is a powerful approach

- easily plugged in various problems/models
- can be used to provide feedback to users
- can embed prior knowledge on label noise

Pointwise probability reinforcements

generic approach to robustify maximum likelihood inference

- many models can be formulated in probabilistic terms
- no parametric assumption $\rightarrow$ could deal with non-uniform noise
- easy to implement if weights can be enforced on instances
- further work: more complex models + noise level estimation

try on your own favorite probabilistic model (and let us know!)

References

- Frénay, B., Verleysen, M. Classification in the Presence of Label Noise: a Survey. IEEE Trans. Neural Networks and Learning Systems, 25(5), 2014, p. 845-869.
- Frénay, B., Kabán A. A Comprehensive Introduction to Label Noise. In Proc. ESANN, Bruges, Belgium, 23-25 April 2014, p. 667-676.
- Frénay, B., de Lannoy, G., Verleysen, M. Label Noise-Tolerant Hidden Markov Models for Segmentation: Application to ECGs. In Proc. ECML-PKDD 2011, p. 455-470.
- Frénay, B., Doquire, G., Verleysen, M. Estimating mutual information for feature selection in the presence of label noise. Computational Statistics & Data Analysis, 71, 832-848, 2014.
- Frenay, B., Hammer, B. Label-noise-tolerant classification for streaming data. In Proc. IJCNN 2017, Anchorage, AK, 14-19 May 2017, p. 1748-1755.
- Frénay, B., Verleysen, M. Pointwise Probability Reinforcements for Robust Statistical Inference. Neural networks, , 50, 124-141, 2014.
- Bion, Q., Frénay, B. Modelling non-uniform label noise to robustify a classifier with application to neural networks. Submitted to ESANN'18 (under review).